

Выводы

Показано, что: 1) лучшим методом оценки векторной близости (среди рассмотренных) является оценка близости по косинусу; 2) метод фильтрации существительных имеет большое практическое значение для классического алгоритма kNN в векторном пространстве, так как, при сохранении только существительных из коллекции документов, точность классификации является максимальной,

одновременно количество используемых терминов для классификации значительно сокращается; 3) с использованием оценки семантической близости точность классификатора kNN значительно увеличивается по сравнению с классической реализацией ($>7\%$), однако требуется дополнительные затраты для вычисления матрицы совместной встречаемости и оценки семантической близости.

СПИСОК ЛИТЕРАТУРЫ

1. Nisbet R., Elder J., Miner G. Handbook of statistical analysis and data mining applications. – Amsterdam: Academic press, 2009. – 824 p.
2. McCallum A., Nigam K. A comparison of event models for naïve bayes text classification // multinomial-aaaiws98. 1998. URL: <http://www.cs.cmu.edu/~knigam/papers/multinomial-aaaiws98.pdf> (дата обращения: 10.01.2012).
3. Joachims T. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization // Joachims_97a. 1997. URL: http://www.cs.cornell.edu/People/tj/publications/joachims_97a.pdf (дата обращения: 10.01.2012).
4. Joachims T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features // Joachims_98a. 1998. URL: http://www.cs.cornell.edu/People/tj/publications/joachims_98a.pdf (дата обращения: 10.01.2012).

5. Handbook of natural language processing / Ed. by N. Indurkha, F.J. Damerau. – London: Chapman & Hall/CRC, 2010. – 692 p.
6. Douglas L.T.R., Laura M.G., David C.P. An improved method for deriving word meaning from lexical co-occurrence // Cognitive Science. – 2009. – V. 7. – № 2. – P. 573–605.
7. Foltz P.W., Kintsch W., Landauer T.K. The measurement of textual coherence with Latent Semantic Analysis // Discourse Processes. – 1998. – V. 25. – № 2. – P. 285–307.
8. Lund K., Burgess C. Producing high-dimensional semantic spaces from lexical co-occurrence // Behavior Research Methods, Instruments and Computers. – 1996. – V. 28. – № 2. – P. 203–208.

Поступила 26.01.2012 г.

УДК 004.931

АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ РАСПОЗНАВАНИЯ ФОРМЫ РУКИ В РЕАЛЬНОМ ВРЕМЕНИ С ИСПОЛЬЗОВАНИЕМ SURF-ДЕСКРИПТОРОВ И НЕЙРОННОЙ СЕТИ

Нгуен Тоан Тханг, В.Г. Спицын

Томский политехнический университет
E-mail: thangngt.cntt@gmail.com; spvg@tpu.ru

Разработан оригинальный алгоритм распознавания формы руки в реальном времени на основе SURF-дескрипторов и нейронной сети. Предложен новый метод генерации дескрипторов для нейронной сети. Создано программное обеспечение для распознавания формы руки в режиме реального времени на основе предложенного алгоритма. Численные эксперименты по распознаванию формы руки на видеопоследовательности в реальном времени показали, что средняя точность распознавания составляет 92 %.

Ключевые слова:

Распознавание руки, SURF-дескриптор, обнаружение признаков, выделение признаков, описание признаков, кластеризация, многослойная нейронная сеть.

Key words:

Hand recognition, SURF-descriptor, feature detection, feature extraction, feature description, clustering, multilayer neural network.

Введение

Распознавание позы и жестов является одной из центральных задач в человеко-машинном взаимодействии и привлекает внимание многих исследователей. Для решения этой проблемы были предложены различные методы и алгоритмы. Они могут быть сгруппированы в две категории: методы на основе внешнего вида руки (*Vision-based approach*), и методы на основе 3D модели руки (*3D hand*

model based approach) [1]. Методы на основе внешности руки используют двумерные признаки изображения для моделирования визуальной внешности руки и сравнивают эти параметры с теми же признаками, выделенными из входного изображения. В методах на основе 3D модели применяют 3D кинематические модели руки, чтобы оценить параметры руки, сравнивая эти параметры с двумерными проекциями 3D моделей. В первую

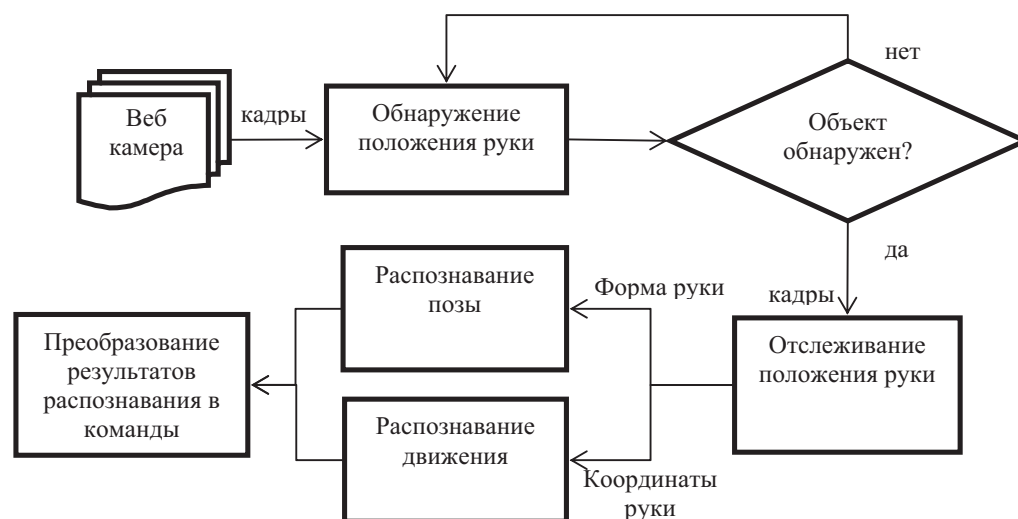


Рис. 1. Алгоритм распознавания жестов на видеопоследовательности

категорию входят методы, в которых используются такие признаки как образы руки [2, 3], локальные признаки [4, 5].

В данной работе предлагается новый алгоритм распознавания жестов на видеопоследовательностях в реальном времени на основе использования SURF-дескрипторов и многослойной нейронной сети. Алгоритм включает следующие этапы (рис. 1): обнаружение объекта (руки) в видеопотоке с помощью детектора Джонса–Виолы, отслеживание положения руки с использованием метода *CAMShift*, выделение характеристических признаков методом *SURF* (*Speeded Up Robust Features*); генерация «словаря» с помощью метода кластеризации *K-means* (*K-means clustering*); создание дескрипторов с совместным использованием *SURF* и *K*-ближайших соседей (*K-nearest neighbors*) на основе созданного «словаря»; обучение нейронной сети с вновь созданными дескрипторами. На основе предложенного алгоритма создан пакет программ, который позволяет распознавать жесты на видеопоследовательностях в реальном времени.

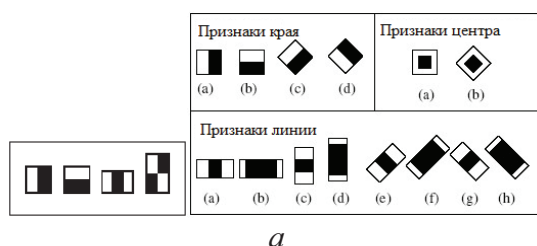
Обнаружение руки в видеопотоке

Обнаружение руки на изображении является первоначальной задачей всех систем распознавания жестов. В настоящее время метод Джонса–Виолы с использованием признаков Хаара и каскадного *AdaBoost* классификатора [6] для обнаружения человеческих лиц считается одним

из лучших алгоритмов для таких задач. Метод Джонса–Виолы был разработан для задачи обнаружения человеческих лиц на изображении. Главное преимущество этого метода состоит в его высокой скорости выполнения и точности обнаружения. В данной работе мы модифицируем детектор Джонса–Виолы с использованием нового набора признаков Хаара для обнаружения руки в видеопотоке в режиме реального времени. В традиционном методе Джонса–Виолы (и в модифицированном *Lienhard* и *Maydt* детекторе) применяется набор признаков, указанный на рис. 2, а. В данной работе мы применяем новый набор признаков, предложенный в работе [7] (рис. 2, б).

Человеческая рука не имеет постоянных контуров, что усложняет задачу обнаружения. Чтобы избежать этой трудности, мы предложили создать два детектора: один – для обнаружения кулака, а второй – для ладони. Кадр из видеокamеры передается сначала в детектор ладони. Если интересующий объект обнаружен, то детектор кулака отключится. В противном случае, детектор кулака включится для обнаружения присутствия руки на кадре.

Для обучения первого детектора (каскадного классификатора *AdaBoost*) мы использовали 1300 позитивных изображений (изображения ладони человеческой руки) размером 20×20 пикселей и 1000 негативных изображений (без объекта). Для второго детектора – 1000 позитивных изображений и 1000 негативных изображений размером 24×24 пикселя.



а



б

Рис. 2. Набор признаков Хаара, используемых: а) в традиционной модели; б) авторами статьи

Отслеживание положения руки в видеопотоке

Методы отслеживания объекта на видеопоследовательностях могут быть разделены на три большие группы: слежение на основе точек (*point tracking*), слежение с использованием ядра (*kernel tracking*) и слежение за контуром (*silhouette tracking*) [8]. Из-за того, что форма руки может резко изменяться на кадрах, слежение на основе точек и контура не подходят для решения данной задачи. В данной работе мы применили метод трекинга на основе метода *CAMShift* с использованием цветowych признаков.

Как только произошел захват руки на кадре, ее положение и размер используются для инициализации поискового окна, содержащего ладонь. Вычисляются две гистограммы: *HM* – гистограмма внутри найденного окна, *HI* – гистограмма целого изображения. Г. Брадски [9] предложил использовать *H*-компонент цветового пространства *HSV* для построения гистограммы. В данной работе цветовые пространства *RGB* и *HSL* используются для сравнения эффективности алгоритма трекинга. Из двух выше построенных гистограмм вычисляется «гистограмма вероятности цветов» (*color probability histogram*) по формуле:

$$P_i = \min \left(\frac{HM_i}{HI_i}, 1 \right). \quad (1)$$

Здесь P_i – i -й компонент гистограммы вероятности цветов; HM_i – компонент гистограммы модели; HI_i – компонент гистограммы целого кадра.

Если заменить каждый пиксель входного изображения соответствующим значением P_i , получим «изображение вероятности» (*probability image*). Для каждого последовательного входного кадра алгоритм повторно вычисляет «центр масс» в обработанной области вокруг предыдущего центра масс в изображении вероятности по формуле (1) и перемещает поисковое окно в это новое положение. Для того, чтобы избежать «вечного» вычисления центра масс, алгоритм прекращает работу после определенной итерации, либо когда изменение центра масс окажется меньше, чем определенный порог:

$$(x_c, y_c) = \left(\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \right),$$

$$M_{00} = \sum_x \sum_y I(x, y),$$

$$M_{10} = \sum_x \sum_y xI(x, y),$$

$$M_{01} = \sum_x \sum_y yI(x, y).$$

Здесь (x_c, y_c) – координаты центра масс; $I(x, y)$ – интенсивность пикселя (x, y) изображения вероятности; M_{00} , M_{10} , M_{01} – нулевой, а также первый момент по направлениям x и y соответственно.

Алгоритм распознавания формы руки

Основная идея предложенного алгоритма состоит в том, что изображение рассматривается как текстовый документ, в котором каждый визуальный признак учитывается как слово, образующее данный документ. Класс документа определяется путем вычисления частоты появления некоторых «ключевых слов». Для применения этой идеи в распознавании объекта вначале из набора изображений (базы данных для обучения) методом *SURF* выделяются характеристические признаки [10]. Затем все выделенные признаки разделяются на группы, используя *K-means* кластеризацию. Каждая группа служит «ключевым словом», из которых строится так называемый «словарь ключевых слов». Deskрипторы для нейронной сети создаются путем сопоставления выделенных признаков входного изображения с ключевыми словами из словаря признаков с помощью поиска *K*-ближайших соседей. Набор deskрипторов и соответствующий класс в дальнейшем используются для обучения нейронной сети. Общая схема алгоритма указана на рис. 3.

Детектор *SURF* базируется на вычислении матрицы Гессена с простой аппроксимацией. Для каждой точки $P=(x, y)$ матрица Гессена $H(P, \sigma)$ на P с масштабом σ определяется по следующей формуле:

$$H(P, \sigma) = \begin{bmatrix} L_{xx}(P, \sigma) & L_{xy}(P, \sigma) \\ L_{xy}(P, \sigma) & L_{yy}(P, \sigma) \end{bmatrix}.$$

Здесь $L_{xx}(P, \sigma)$, $L_{yy}(P, \sigma)$, $L_{xy}(P, \sigma)$ – свертка второй производной Гаусса $\frac{\partial^2}{\partial x^2} g(\sigma)$, $\frac{\partial^2}{\partial y^2} g(\sigma)$,

$\frac{\partial^2}{\partial x \partial y} g(\sigma)$ с изображением в точке P , соответственно.

С целью сокращения времени обработки, вторая производная Гаусса приближенно вычисляется с использованием «*box filter*» и «интегрального изображения». Интегральное изображение также упрощает процесс анализа в пространстве масштабов. Детерминант матрицы Гессена затем используется для выбора координат точек интереса и масштаба. Точки интереса локализуются с помощью «подавления не-максимумов» (*non-maximum suppression*). Это делает *SURF* признаки устойчивыми к масштабированию. *SURF* также строит окружность вокруг выделенной точки интереса, чтобы определить уникальную ориентацию, и тем самым дает *SURF* признакам инвариантность к вращению. *SURF* признаки описываются векторами, длина которых фиксирована (64 в оригинальном *SURF* и 128 в расширенной версии *SURF*).

В связи с тем, что *SURF* выделяет точки интереса на основе локальных признаков, количество обнаруженных точек не фиксировано, а меняется

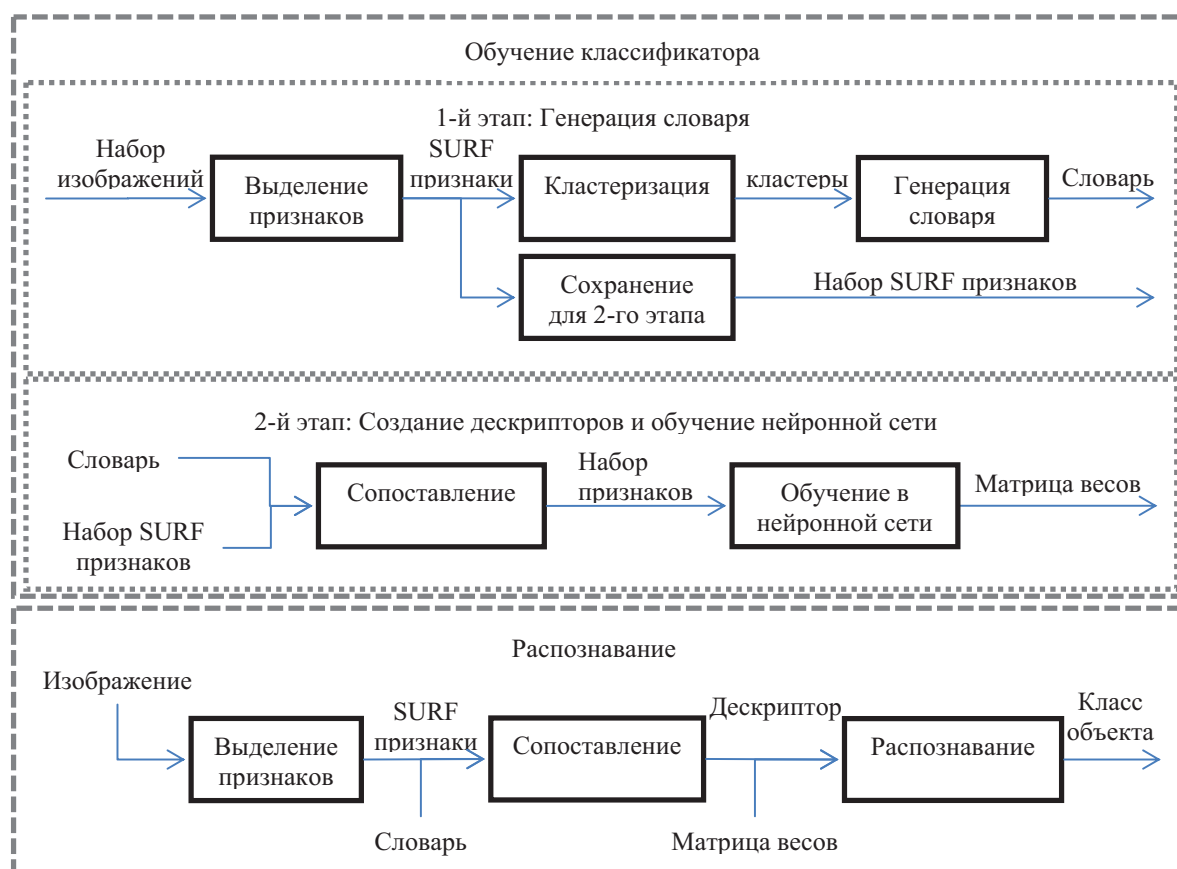


Рис. 3. Общая схема алгоритма распознавания формы руки

в зависимости от изображения, даже для одного и того же объекта. Поэтому нельзя напрямую подавать данные дескрипторы в нейронную сеть для распознавания. Для решения такой проблемы предлагается разделить все обнаруженные признаки из набора изображений на группы. Это обусловлено тем, что визуальные признаки, выделенные из одного класса объекта должны иметь сходство. Признаки, полученные при анализе одного изображения, также могут быть похожи друг на друга. Разделение признаков на группы позволяет «соединить» подобные признаки вместе и выбрать одного «представителя» для всей группы. Описание объекта тогда можно осуществлять с помощью только этих «представителей». В данной работе используется популярный метод кластеризации *K-means*.

Метод *K-means* разбивает множество элементов векторного пространства на заранее известное число кластеров k . Основная идея заключается в том, что на каждой итерации перевычисляется центр масс для каждого кластера, полученного на предыдущем шаге, затем векторы разбиваются на кластеры вновь в соответствии с тем, какой из новых центров оказался ближе по выбранной метрике. Алгоритм завершается, когда на какой-то итерации не происходит изменения кластеров. Это происходит за конечное число итераций, так как количество возможных разбиений конечного множества конечно, причем на каждом шаге суммарное

квадратичное отклонение уменьшается, поэтому заикливание невозможно. *K-means* стремится минимизировать суммарное квадратичное отклонение точек кластеров от центров этих кластеров:

$$V = \sum_{i=1}^k \sum_{x_j \in S_i} (x_j - \mu_i)^2.$$

Здесь k – число кластеров; S_i – полученные кластеры; μ_i – центры масс векторов $x_j \in S_i$; x – набор входных векторов.

При применении *K-means* кластеризации нужно вначале задать количество кластеров. Если кластером считается множество однозначных слов (как в текстовом документе), то количество кластеров – это общее число «ключевых слов». Центром масс каждого кластера является само ключевое слово, а совокупность всех полученных ключевых слов составляет словарь. Если считается, что выделенные *SURF* методом признаки из объектов определенного класса уникальны и характерны только для данного класса объекта, то количество слов в словаре (число кластеров) приблизительно равняется сумме средних чисел признаков всех классов объекта:

$$K = \sum_{i=1}^N \frac{\sum_{j=1}^{M_i} k_j}{M_i}.$$

Здесь K – число кластеров; N – количество классов объектов; M_i – количество образцов i -го класса; k_j – число выделенных признаков из j -го образца i -го класса.

При завершении процесса кластеризации получается набор векторов центра масс всех групп *SURF*-дескрипторов. Сами векторы центра масс являются *SURF*-дескрипторами и могут служить «представителем» для всех векторов в своей группе. Набор векторов центра масс (словарь) используется для генерации новых дескрипторов для классификатора.

При поступлении нового изображения, *SURF* выделяет характеристические признаки и описывает эти признаки набором векторов, размерность которых фиксирована. Для каждого *SURF* вектора (дескриптора) вычисляется квадрат расстояния Евклида до каждого из центров масс по формуле (2) и выбирается самое короткое расстояние.

$$d(p, q_k) = \sum_{i=1}^N (q_{ki} - p_i)^2. \quad (2)$$

Здесь $d(p, q_k)$ – расстояние от входного *SURF* вектора до k -го центра масс; N – размерность *SURF*-вектора (64 или 128 в зависимости от выбранного способа описания); p – входной *SURF* вектор; q_k – k -й центр масс.

Если значение этого расстояния меньше чем определенный порог, то соответствующий центр выбирается в качестве представителя для поступающего *SURF*-вектора. Другими словами, данный *SURF*-вектор заменяется соответствующим центром масс. При создании базы данных для генерации словаря используются изображения без фона, чтобы выделенные признаки были характерны только для объектов. Однако на изображениях в базе данных для обучения нейронной сети могут присутствовать фон или другие объекты. Поэтому нужно установить пороговое значение расстояния, чтобы исключить «чужие признаки», которые выделяются, к примеру, из фона, а не из объекта. Этот порог определяется как среднее значение всех расстояний между центрами масс:

$$\partial = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N d(q_i, q_j), i \neq j.$$

Здесь ∂ – пороговое значение; $d(q_i, q_j)$ – расстояние между центрами q_i и q_j ; N – число центров.

Когда все входные *SURF*-векторы данного изображения заменяются и обозначаются соответствующими «словами», проводится построение гистограммы, в которой указывается частота появления каждого ключевого слова из словаря. Эту гистограмму будем называть «*BOW*-дескриптором». Для каждого объекта *BOW*-дескриптор имеет фиксированную длину, равную числу слов (кластеров) в словаре признаков. Следует отметить, что *BOW*-дескриптор принимает значения, которые характерны для каждого класса объекта, и, таким образом, указанные значения идеально подходят для нейронной сети.

Поскольку *SURF* выделяет локальные признаки, не учитывая относительные связи между ними, и в гистограмме вычисляется частота появления определенных *SURF* признаков, пространственные отношения между признаками объекта также не учитываются. В распознавании руки, данное свойство *BOW*-дескрипторов очень полезно, так как нужно обучать нейронную сеть только один раз, скажем, для распознавания левой руки. Существенным является тот факт, что при тестировании нейронная сеть оказывается способной распознавать еще и правую руку. Таким образом, мы получаем возможность распознавать обе руки одновременно в одном классификаторе.

При разработке алгоритма для распознавания формы руки в режиме реального времени скорость работы классификатора является приоритетным показателем. Многослойная нейронная сеть удовлетворяет этому требованию. Данный тип нейронной сети также обеспечивает высокую точность распознавания. Кроме того, *BOW*-дескрипторы с низкой размерностью и фиксированной длиной идеально подходят для нейронной сети. Поэтому в данной работе в качестве классификатора применяется многослойная нейронная сеть с обратным распространением ошибки.

Используемая нейронная сеть в предложенном алгоритме состоит из N входных нейронов, где N – количество слов в словаре, K выходных нейронов, где K – число классов. Количество нейронов в скрытом слое определяется эмпирическим способом. В качестве функции активизации применяется симметричная биполярная сигмоидальная функция:

$$f(x) = \frac{\beta(1 - e^{-\alpha x})}{(1 + e^{-\alpha x})}.$$

Здесь α и β – свободные параметры.

Для повышения скорости сходимости сети применяются модификации: *Nguyen-Widrow* инициализация, моментум, и групповое обновление.

Реализация и результаты экспериментов

В работе реализована система распознавания четырех классов позы руки (рис. 4) с помощью предложенного алгоритма.

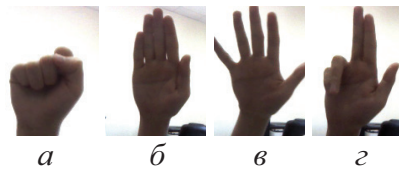


Рис. 4. Классы используемых объектов в программе: а) камень; б) бумага; в) открытая бумага; г) ножницы

На этапе генерации «словаря» для каждого класса отбираются по 25 «чистых» изображений (без фона) с объектом, снятым в различных ракурсах и масштабах. Количество выделенных признаков зависит от значения порога Гессена. Если установить маленькое значение порога, то количество признаков

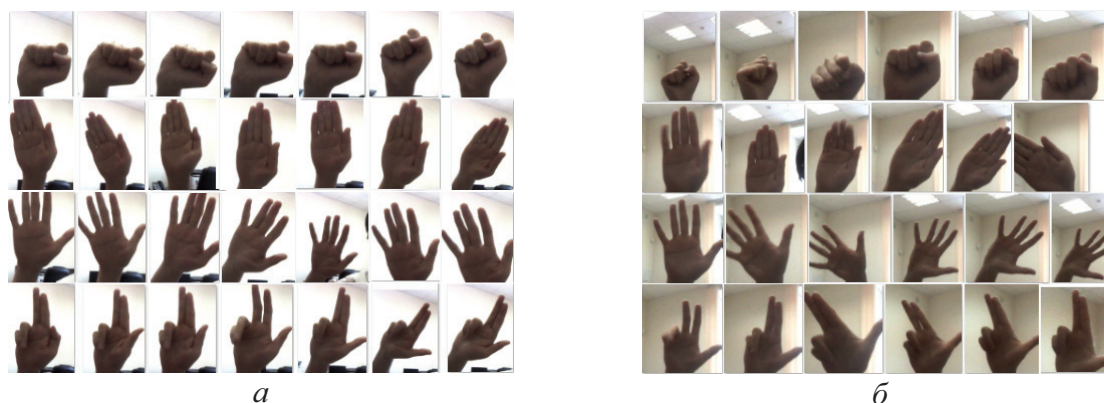


Рис. 5. Изображения: а) часть базы данных для обучения; б) тестовые

увеличивается, но при этом требуется длительное время обработки. Если установить большой порог, то число выделенных признаков может быть не выгодным для распознавания. Для изображения заданного размера порог Гессена был выбран таким образом, чтобы для изображений класса (а) *SURF* выделял в среднем 30 признаков; для класса (б) – 70; для класса (в) – 90; для класса (г) – 80. В этом случае число слов в словаре равно 270.

Для обучения нейронной сети в базу данных добавляются еще 75 изображений для каждого класса. Изображения для обучения были сняты на простом фоне для повышения робастности классификатора (рис. 5, а). Для тестирования классификатора веб-камерой были сняты 1000 изображений для каждого класса (рис. 5, б). Размер каждого изображения составляет 100×100 пикселей.

Программа для апробации работы алгоритма была реализована на языке C#. Программа предоставляет возможность генерации словаря признаков, формирования базы данных для обучения, создания и обучения нейронной сети, а также име-

ет отдельный интерфейс для проверки и тестирования работоспособности. Тестирования были выполнены на ноутбуке *ASUS UL VT80* с операционной системой *Windows 7 64 bit* и встроенной веб-камерой. Результаты тестирования выборки из 1000 различных изображений при времени обработки одного кадра порядка 60 мс показывают: камень – правильных ответов 874; бумага – 912; открытая бумага – 945; ножницы – 932.

Таким образом, средняя точность распознавания составляет ~92 %. Самая высокая точность получается при распознавании «открытой бумаги». Это можно объяснить тем, что для «открытой бумаги» число выделенных признаков больше всех. Для «камня» количество признаков намного меньше остальных. Это и объясняет низкую точность распознавания класса «камней».

Время обработки для всех изображений составляет ~60 мс, что позволяет работать в реальном времени со скоростью около 15 кадров в секунду. На рис. 6 представлены результаты работы программы в реальном времени.



Рис. 6. Распознавание руки в реальном времени

Выводы

Предложен и реализован новый алгоритм для распознавания руки на основе *SURF*-дескрипторов и нейронной сети. Изложен новый метод генерации дескрипторов для нейронной сети, которые устойчивы к поворотам и масштабированию, частично к изменению освещения, и робастны на сложном фоне. Созданные дескрипторы позволяют распознавать левую и правую руки одновременно, используя только один классификатор, при этом обучение требуется только один раз для одной руки. Алгоритм

позволяет распознавать форму руки с высокой точностью. На основе предложенного алгоритма реализовано программное обеспечение для распознавания формы руки в видеопотоке в режиме реального времени. Программа поддерживает генерацию «словаря» признаков, формирование базы данных для обучения, создание и обучение нейронной сети. Проведенные численные эксперименты по распознаванию формы руки на видеопоследовательности в реальном времени показали, что средняя точность распознавания составляет 92 %.

СПИСОК ЛИТЕРАТУРЫ

1. Zhou H., Huang T.S. Tracking articulated hand motion with Eigen dynamics analysis // Proc. of Intern. Conf. on Computer Vision. – Nice (France), 2003. – V. 2. – P. 1102–1109.
2. Chen F., Fu C., Huang C. Hand gesture recognition using a real-time tracking method and Hidden Markov Models // Image and Vision Computing. – Oxford: Elsevier, 2003. – V. 21. – № 8. – P. 745–758.
3. Ng C.W., Ranganath S. Gesture recognition via pose classification // Proc. 15th Intern. Conf. on Pattern Recognition. – Barcelona (Spain), 2000. – V. 3. – P. 699–704.
4. Oka K., Sato Y., Koike H. Real-time fingertip tracking and gesture recognition // Proc. IEEE Computer Graphics and Applications. – 2002. – V. 22. – № 6. – P. 64–71.
5. Huang C., Jeng S. A model-based hand gesture recognition system // Machine Vision and Application. – 2001. – V. 12. – № 5. – P. 243–258.
6. Viola P., Jones M. Rapid object detection using a boosted cascade of simple features // IEEE CVPR01. – 2001. – P. 511–518.
7. Kolsch M., Turk M. Robust hand detection // Proc. of the 6th IEEE Intern. Conf. on Automatic Face and Gesture Recognition. – Seoul (Korea), May 2004. – Seoul, 2004. – P. 614–619.
8. Alper Y., Omar J., Mubarak S. Object tracking: A Survey // ACM Computing Surveys (CSUR). – 2006. – V. 38. – Iss. 4. – Art. № 13.
9. Bradski G.R. Real time face and object tracking as a component of a perceptual user interface // Proc. of the 4th IEEE Workshop on Applications of Computer Vision. – New Jersey (USA), October 1998. – New Jersey, 1998. – P. 214–219.
10. Bay H., Ess A., Tuytelaars T., Van Gool L. SURF: Speeded Up Robust Features // Computer Vision and Image Understanding. – Oxford: Elsevier, 2008. – V. 110. – № 3. – P. 346–359.

Поступила 11.03.2012 г.

УДК 004.932

РАСПОЗНАВАНИЕ ЛИЦ НА ОСНОВЕ ПРИМЕНЕНИЯ МЕТОДА ВИОЛЫ–ДЖОНСА, ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЯ И МЕТОДА ГЛАВНЫХ КОМПОНЕНТ

Буй Тхи Тху Чанг, Фан Нгок Хоанг, В.Г. Спицын

Томский политехнический университет
E-mail: trangbt.084@gmail.com

На основе совместного применения метода Виолы–Джонса, вейвлет-преобразования и метода главных компонент предложен новый алгоритм распознавания лиц на цифровых изображениях и видеопоследовательностях в режиме реального времени. Описан алгоритм и разработано программное обеспечение для распознавания лиц. Приведен пример работы и представлены результаты тестирования программы. Показано, что использование предложенного оригинального алгоритма дает возможность эффективного распознавания лиц на цифровых изображениях и видеопоследовательностях.

Ключевые слова:

Распознавание лиц, метод Виолы–Джонса, метод главных компонент, вейвлет-преобразование.

Key words:

Face recognition, method Viola-Jones, principal component analysis, wavelet transform.

Введение

Распознавание лиц является одной из самых изученных задач в таких областях как цифровая обработка изображений, компьютерное зрение, биометрия, организация видеоконференций, создание интеллектуальных систем безопасности и контроля доступа и т. п. Процесс распознавания лиц

обычно состоит из двух этапов: поиск области лица на изображении, и сравнение найденного лица с лицами, находящимися в базе данных. В настоящее время метод Виолы–Джонса [1] является самым популярным методом для поиска области лица на изображении из-за его высокой скорости и эффективности. Детектор лица Виолы–Джонса